Prospector: An adaptive front-end to the Google search engine

Christian Schwendtner, Florian König, Alexandros Paramythis
Johannes Kepler University
Institute for Information Processing and Microprocessor Technology (FIM)
Altenbergerstraße 69, A-4040 Linz, Austria
{schwendtner, koenig, alpar}@fim.uni-linz.ac.at

Abstract

This paper presents Prospector a front-end to the Google search engine which, using individual and group models of users' interests, re-ranks search results to better suit the user's inferred needs. The paper outlines the motivation behind the development of the system, describes its adaptive components, and discusses the lessons learned thus far, as well as the work planned for the future.

1 Introduction

1.1 Background and motivation

The motivation underlying the Prospector's development has been to investigate the extent to which a simple yet effective adaptive meta-search layer can be applied as a front-end to a popular search engine, for personalizing search results. The idea of personalizing web search results is not new, but is increasingly relevant as the sheer number of pages / sites available online rises at immense paces, with synonimity and homonimity compounding the problems of identifying search results that are truly relevant to the user's search [Tanudjaja and Mui, 2002].

The feasibility of creating such a personalization layer has been largely dependent on the availability of public services exposing an API for accessing the search functionality of major search engines. The development of the Prospector commenced in the spring of 2005, at which time we chose to implement it as a meta-layer to the Google search engine, due to the maturity of the respective API[1], and the overall standing of Google as the most popular search engine in the world. The Prospector utilizes Open Directory Project (ODP)[2] metadata for effecting user- and group-oriented re-ranking "on top" of the original search results. This data was originally provided by the Google Search API, and, in the current version of the system, is derived directly from the ODP site.

The design goals we had on the outset can be summarized as follows:

- The adaptation algorithm should be as simple as possible, and should be based on the users' interests (both at the individual- and group- levels), as these are inferred by characterizing search results, and identifying their thematic classification(s) within an established taxonomy. Although simple, the algorithm should be capable of supporting, directly or as extensions, standard features of similar systems, such as "aging" of user interests (see, e.g., [Koychev and Schwab, 2000]).
- Along the same lines as the adaptation algorithm, the derivation of the user and group models should be as simple as possible. Input to the modeling process was to include: (a) implicit ratings of search results, inferred from specific types of user behaviour (such as marking a result link as unsuitable, even without following it); and (b) explicit ratings of individual results by users.
- The personalization features of the Prospector, at the individual level, were to be available to users as an optional feature that requires registration (and, consequently, logging into the system prior to issuing search queries). Nevertheless, it was desirable to utilize group models to provide a "generic" level of adaptivity to non-registered users as well, on the basis of the thematic categories mentioned earlier.
- Last but not least, the Prospector was to provide a scrutable user model [Kay, 2000], enabling registered users to both inspect their user model, and modify it with respect to their interests.

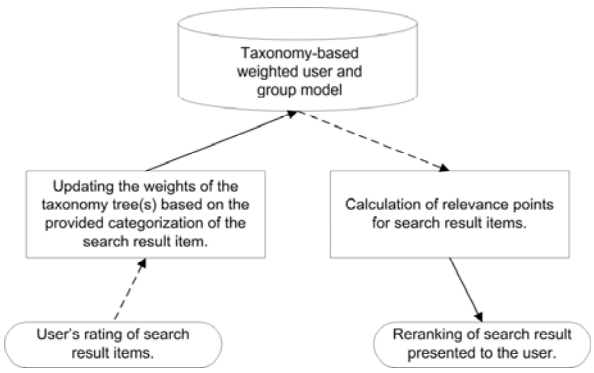

Figure 1: A synoptic view of adaptivity in the Prospector.

Using the categorization scheme introduced by Jameson [2003], the intended adaptive features of the

[1] For more information see: http://www.google.com/apis/
[2] Open Directory Project: http://dmoz.org/

Prospector could then be roughly summarized as depicted in Figure 1.

1.2 Related work

The Prospector can be broadly categorized as an adaptive information retrieval system. The literature on this type of systems is too extensive to cover here; interested readers may refer to [Micarelli and Sciarrone, 2004], and [Pierrakos *et al.*, 2003] for related work on adaptive information filtering and Web personalization.

A representative and widely acclaimed system that we will use as an example is I-SPY [Smyth *et al.*, 2003a; 2003b]. I-SPY implements an adaptive collaborative search technique that enables it to selectively re-rank search results according to the learned preferences of a community of users. Effectively I-SPY actively promotes results that have been previously favored by community members during related searches so that the most relevant results are top of the result list [Smyth *et al.*, 2003a]. I-SPY monitors user selections or "hits" for a query and builds a model of query-page relevance based on the probability that a given page will be selected by the user when returned as a result to a specific query.

Google has itself introduced two versions of personalized search functionality. The first version, unveiled in March 2004, allowed users to create a profile used to customize search results. By selecting categories, one could tell Google that they are interested in things like movies, radio and music. Then by using a slider, users could "personalize" their results to skew them toward their particular interest areas. This first version was based on classification of pages across the web into topics. The "personal" results were those skewed more toward the topics areas users were interested in, according to the profile they manually created.

The current incarnation of the service[3], follows an entirely different approach. Although details of the algorithms used have not been published to date, the general principles are as follows: (a) User profiles are built up by monitoring the user's search behaviour, as well as the links the user follows among the search results. (b) The only way in which users can modify their profiles is by removing items from their personal search history, which is where all information about past user behaviour is stored. (c) When this service is applied (although it is not clear under what conditions it is triggered), the search results are re-ranked to better suit the user's profile. When this happens, a link is also provided to allow the user to see the results in unmodified order.

The systems most relevant to the Prospector are the ones described in [Tanudjaja and Mui, 2002], and [Chirita *et al.*, 2005]. The first paper describes Persona, a system which utilizes ODP metadata for creating taxonomies of user interests and disinterests and tree coloring to represent user profiles. Taxonomy nodes visited are 'colored' by the number of times they have been visited, by user ratings if available, and by the URLs associated with the node [Tanudjaja and Mui, 2002]. The second paper describes a system with more similarities to the Prospector. Specifically, Chirita *et al.* [2005] have used ODP metadata to create user profiles, and then used various approaches to calculating the distance between a given search result and the user's profile, to decide that result's rank. Users pre-select ODP categories that they are interested in, for the creation of their profiles; the system does not have an adaptive component, so these profiles do not evolve over time. The distance calculation approaches range from what the authors term "naïve" –based primarily on graph node distances–, to a version of the PageRank [Brin and Page, 1998] algorithm modified to include a measure of the semantic similarity between nodes in an ODP taxonomy. User-based experiments have shown that these approaches to search personalization deliver superior results to their non-personalized counterparts [Chirita *et al.*, 2005].

The Prospector shares some of the characteristics, but also differs in several ways from the approaches outlined above. In synthesis, the Prospector: is intended as a meta-layer or front-end to a search engine; creates and utilizes user models built from explicit user ratings of items; represents user (dis-) interests using ODP-based thematic taxonomies; maintains (thematic) group models, and uses them in conjunction with the individual user model to adapt search results; and, supports user model scrutability.

2 The Prospector system

As already mentioned, the Prospector is a front-end to the Google search engine. A basic anonymous search returns exactly the same results as a search made directly on the Google site. Adaptivity comes into play in two guises: firstly, while still remaining anonymous, the user can have the results re-ranked by the system, according to thematically-based group profiles; and, secondly, users can register (and log in), progressively building up their personal interest profile, which is then used to automatically re-rank search results. The rest of this section outlines the overall functionality of the system, and then goes on to discuss the modeling and adaptivity aspects of the system in more detail.

2.1 Overall functionality

Anonymous searching

The main page of the Prospector is depicted in Figure 2. Apart from the search field itself, there is a set of links at the top of the page which allow users to register and log in, as well as to acquire additional information about the system and how it works. Search results appear under the search pane (gray area in Figure 2), as shown in Figure 3.

Figure 2: The main page of the Prospector.

Note that, above the search results, there is a form which lets one re-rank the search results according to the interest profile of a specific group (groups are organized according to thematic categories; more details are provided in the next section). When a re-ranking has been performed, the group whose profile was used for re-ranking is indicated under the form (see Figure 4). Also note that, whenever re-ranking takes place, the Prospector

[3] Available at: http://www.google.com/psearch

places a relevance score expressed as points next to each item. The derivation and significance of this score is explained in the next section.

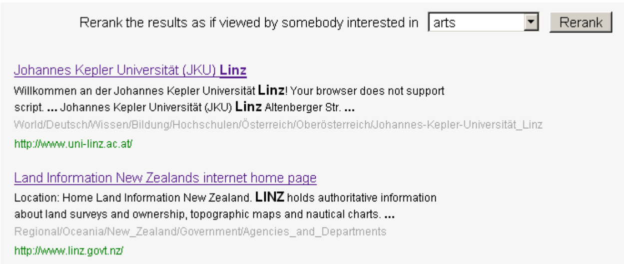

Figure 3: Unranked search results, as returned by Google.

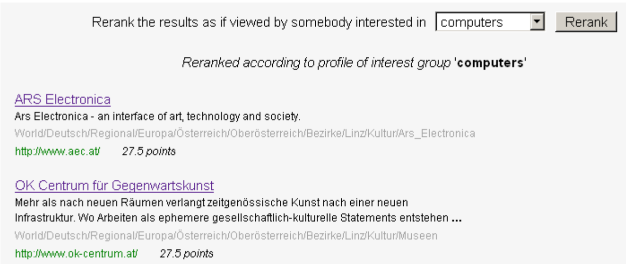

Figure 4: Anonymous search results re-ranked according to the profile of the thematic group 'computers'.

User-aware searching

User-aware searching is available to users that are registered and logged in. This results in search results being automatically reordered according to the user's interest profile, and associated user model. Both of the later are accessible through respective entries in the link bar.

Typically, the first task of registered users that log in for the first time is to specify their interests. This is done through a form that contains a listing of selected top-level categories of Google's thematic taxonomy. Users are able to indicate that they have no interest, or use a 5-point scale to rate their interest, in a particular theme (see Figure 6). Users can return to this page at any time to adjust their entries.

From the available choices on this form, 'No interest' neither influences the ranking of results, nor alters the group profile when rating. The higher the extent of a user's interest in a group, the more the profile influences the reordering and the more a rating influences the group's profile.

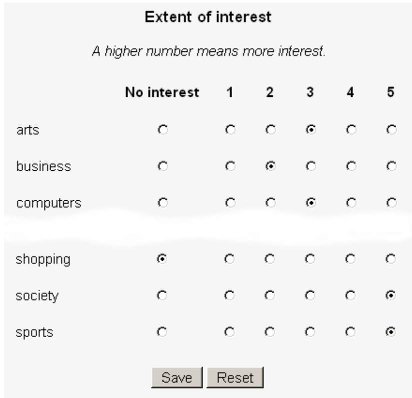

Figure 6: Specifying user interests (interest groups).

User-aware searching is done in much the same way as anonymous searching, as far as the interactive part of the system is concerned (see Figure 7). There are two noteworthy exceptions however: as a logged-in user one can disable a group-oriented reordering and rank the search result items according to the combined user and group profiles (using the link '[disable]' in Figure 7); and, the system adds a link after each result item which allows users to both remove the item from the list displayed and, at the same time, rate it negatively (this is the graphical link with the caption 'Unsuitable' in Figure 7).

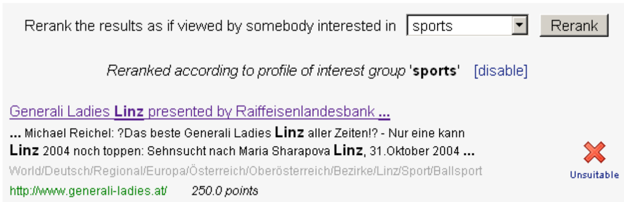

Figure 7: Search results for logged in users.

When a user follows a link among the search results, the respective page is shown with a rating frame above it (see Figure 7). Users can then rate a page positively ('Result OK') or negatively ('Result NOT OK') and can also choose whether to return to the search results display ('Take me back') or simply remove the rating frame ('Stop searching'). Rating a page negatively and returning to the

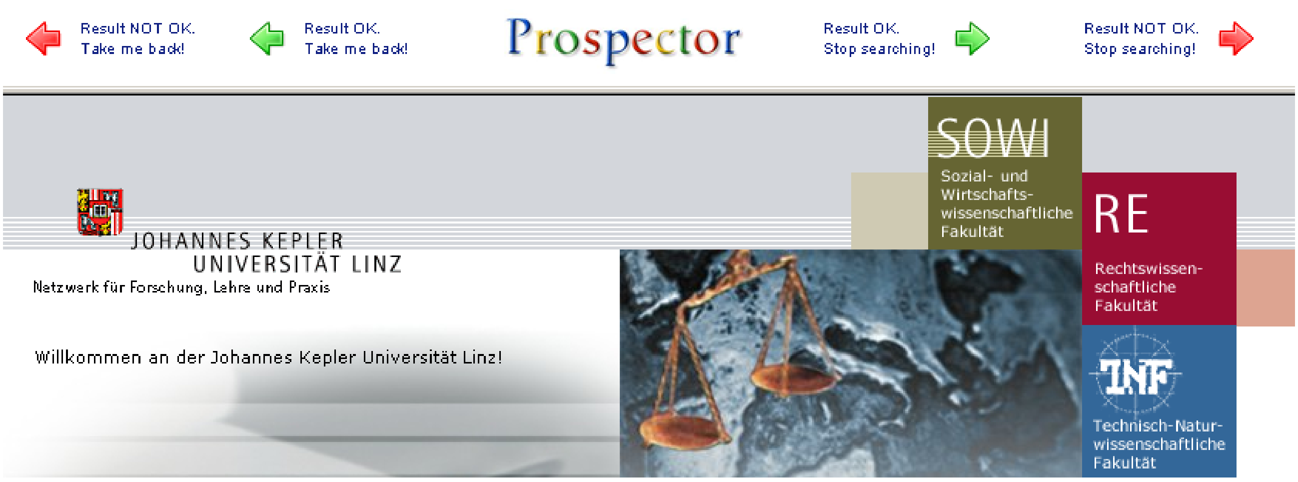

Figure 5: Rating frame above the actual result page.

search results automatically removes that page from the list of items.

The Prospector's interface also provides support for viewing and modifying one's own user model (following the "scrutability" principles of user modeling). This feature will be discussed in the next section, as an understanding of what is modeled and how is necessary for understanding that part of the system.

2.2 Modelling and adaptation

User and group modeling

Modelling of user and group interests is done using a thematic taxonomy. This taxonomy is derived from the hierarchical classification scheme provided by the Open Directory Project. In the first incarnation of Prospector, search results retrieved programmatically through the Google search API were annotated with a 'path' that identified the page's categorization within that hierarchy[4]. These paths formed the basis for user and groups models. The top-level categories in this scheme were the ones used to enable users to express their general thematic interests, as described in the previous section.

This approach had to be modified somewhat, when the ODP category stopped being included in the results returned by the Google search API[5]. There are, in general, two alternative ways in utilizing ODP metadata. The first is to use the data that ODP makes freely available from their site, and construct a layer / component that allows for querying that data directly. Although this approach requires custom development efforts, and makes for a quite heavy-weight system (the data alone is several hundred MBs *compressed*), one might argue that it is also the most cohesive and self-sufficient. An alternative approach, used as an intermediate solution in Prospector, is to utilize the ODP search functionality already available online. In short, this involves using a URL as the query string, and receiving as results the ODP categories (i.e., taxonomy nodes) under which the given URL has been classified. This solution can, of course, only serve as a temporary one, since it introduces a significant delay in the search, by requiring a request-response cycle to identify the ODP category (or categories) of each result returned by the underlying search engine.

Two types of user activity are taken into account for updating the user and group models: the user's removing an item from the search results as unsuitable; and, the user's providing an explicit, positive or negative, rating on a page after having visited it.

When either type of activity is encountered, the following take place, as far as the individual user model is concerned: if the 'path' corresponding to the rated item does not exist in the user model, it is added; for each node on the path the weight is changed to reflect the user's rating. Weight is subtracted or added to reflect a negative or, respectively, positive rating. The exact change in the weight in each node is affected by the following factors: (a) the "depth" of the node – more specific nodes are affected the most; and, (b) the user's interest ranking for the top-level category of the path – paths of higher interest to the user are affected more.

Group modeling in the Prospector functions along the same premises, and can be thought of as using predefined thematic clustering, with users "belonging" to different clusters with varying degrees of affinity (based on their self-expressed interest rankings). In other words, the Prospector maintains a group model for each of the top-level thematic categories. Apparently, this results in each group model representing a distinct portion of the overall taxonomy.

Updating of the group models occurs whenever there is an update in the model of an individual user that has any degree of affinity to the group. To start with, for each such change, paths are added as necessary. Subsequently, the changes in the individual user model are propagated to the group model. The impact of the weight propagations is itself weighted using the degree of affinity of the user to the group.

Adaptive reordering of search results

The primary adaptive function of Prospector is the reordering / re-ranking of search results. This is done using the current user's individual interest model, combined to varying degrees with the models of groups to which the user has some degree of affinity. Specifically, the models are used to calculate a relevance score for each item, which is in turn used to reorder items bringing the potentially most relevant ones to the top.

Let's take a closer look at the process of calculating the relevance score ("points") for a single search result. Assume that a single result item of a Google search belongs to the category "World / Sports / Basketball". To calculate the points, the categories of the search result item are retrieved, and the algorithm looks for the corresponding path through the portion of the taxonomy already represented in the model. So in our example we try to find a root element of the tree named "World". If it is found, the points assigned to the respective tree node are retrieved, and the algorithm continues with the next category entry ("Sports"), which it looks for among the children of the current tree node. If it is found, the points assigned to the tree node are added to the current sum. This process is repeated until the entire path is covered, or there is no node corresponding to a given path fragment.

As already mentioned the Prospector allows users to specify their interest in (or, degree of affinity to) a group, using a 5-point scale. This "degree" is used to repeat the above described process using the models of each of the groups to which the user belongs. The result of this process is that the ranking of each item is influenced by the ranking of other group members for the category under which the item is classified.

In practice, the ranking algorithm does not semantically distinguish between group models and the user's personal model. Instead, it receives a list of models and a list of weights that define the "importance" of each. The highest weight is assigned to the model corresponding to the user, which renders it the primary factor in determining an item's rating, but still leaves plenty of room for benefiting from groups ratings on categories that the user has not rated yet (thus addressing the "bootstrapping" problem for individual models).

Individual models can both be inspected and modified by users using forms like the one shown in Figure 8. Tree nodes with a positive weight represent categories the user

[4] To be precise, this was actually the path in the hierarchy maintained under Google's own directory service (http://directory.google.com), which, however, is practically identical to the ODP, as it is based on the same data.
[5] This was a decision on the part of Google, who announced that this change is to be considered a permanent one.

appears to be interested in according to the ratings performed, with negatively rated nodes represent uninteresting categories. Nodes with a weight of 0.0 don't contribute to the ranking of a result item. The weights can be changed to directly alter the user profile. When checking the box in the column 'Disable sub-tree' the corresponding node and all its sub-nodes are set to a weight of 0.0.

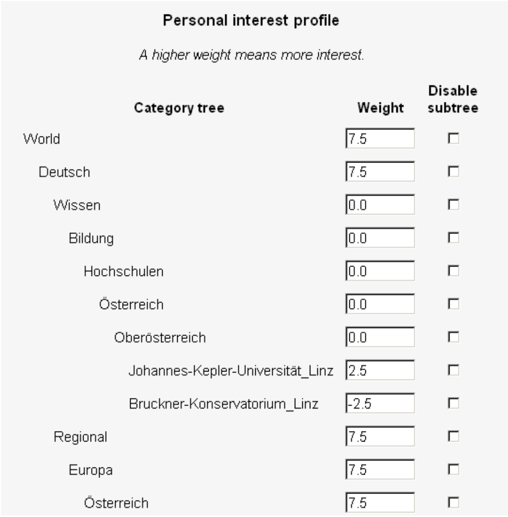

Figure 8: Viewing and modifying the user model.

3 Discussion

Preliminary evaluation

To evaluate the system, we have engaged in two types of preliminary evaluation activities. Firstly, we conducted a informal, usability-oriented heuristic evaluation with the assistance of affiliated usability experts. Secondly, we asked members of our institute to use the system for the period of one day, and asked them to provide their feedback on an individual basis, without prescribing the type or range of input we were looking for.

Very important technical limiting factors which constrained the types of assessment feasible at this stage, and may have also influenced the results, were that each search query returned only 50 results (in 5 pages of 10 results each), and took considerably longer than it would have done, had it been issued directly on the Google site. Both of these constraints were due to limitations in the Google API, which returns only 10 result items per query request. We decided to perform 5 such requests, to have a sufficient amount of links to work with. This however meant that a normal search, from the perspective of the user, took approximately 5 times as long as they might have expected. Experts and users participating in the studies had been advised regarding these limitations, and were asked to try and disregard them as much as possible.

The findings of this preliminary informal round can be summarized as follows:

- Users were in general positively disposed towards, and rather satisfied with the effects of reordering. It should be noted however, that there were no control settings, and users had no way of comparing the reordered results to the ones they would have gotten directly from Google, other than repeating the search on the Google site. Furthermore, we anticipate that the results would have been even better, if the system was used over a longer period of time, thus having more opportunity to build more comprehensive user and group models.
- The relation between the top-level categories (used to rank user interests in different themes) and the user model representation (where categories are associated with weights) was found to be confusing. Specifically, users found it hard to anticipate how changes in one part of the system affected the other. They were also unclear about the effects of weights on the search results.
- Another source of confusion was the use and exposure of arbitrary weights for items. The way the modeling and adaptation algorithms work at the moment results in weights that may vary significantly in range. Although this has beneficial effects on the ranking, it is apparently not easy to comprehend. It also presents an interactivity challenge, as users have to guess what weights might be appropriate for nodes in their model, without having any semantic interpretation of the available ranges.

Planned improvements

Based on these preliminary results, we have planned a number of improvements to be made to the system. These include:

- The unification of the top-level categories used to derive the initial user profile, with the contents of the user model. Specifically, we intend to eliminate the differentiation between the two, and simply use the initial user input to seed the user model. Along the same lines, we will also eliminate the “Interests” view and use the user model view as the only one from where the user’s model can be inspected and modified.
- The modelling and adaptation algorithms will be modified to use a probabilistic approach to weighting, instead of the current unconstrained one. Specifically, we intend to apply two correlated types of normalization in the derivation of weights for items, and in the modifications of node weights on the basis of user ratings. This may have a somewhat negative impact on the ranking results, but will provide for a concise and easy to comprehend range of values in the models.
- Following from the previous step, we intend to modify the representation of weights in the user model view, replacing the weight text field with a pseudo-continuous scale that can be adjusted by means of a slider control.
- Finally, an additional option will be added to the interface for logged in users, which will allow them to view the search results without any ranking applied to them. Although this feature is expected to be of limited utility in the general case, it is expected to have at least two important benefits: firstly, it may prove useful in building up user trust in the system’s adaptive behaviour; secondly, it may serve as a “backup” option for cases where wrongly inferred

(dis-) interests at the user- or group- levels skew results in an undesirable direction.

Further to the above, we intend to release the Prospector as open source software. We feel that the modularity of the system render it a potentially interesting "playground" for adaptive / personalized search work, or, alternatively, a viable tool for hands-on work in adaptive hypermedia courses. Fr instance, Prospector allows, among other things, for: easy migration to different search engines; the implementation of alternative modeling and adaptation algorithms; the modification of the user interface, independently of the adaptive core; etc.

Planned evaluation activities

Once the improvements described in the previous section are in place, Prospector will undergo two evaluations.

- The first will be performed in October 2006 by the Department of Technical and Professional Communication, Faculty of Behavioural Sciences, University of Twente (the Netherlands), and will have a strong focus on the user. In this study, Prospector will be evaluated as an adaptive system *per se*, but also used as a case study for determining the appropriateness of different evaluation methods in assessing how users experience adaptivity.
- In a second stage we intend to engage in a full-scale evaluation of the system, using the layered evaluation framework for adaptive systems, introduced by Paramythis and Weibelzahl [2005]. In fact, we intend to approach the formal assessment of the system from two different directions: On the hand, we will use the aforementioned evaluation framework to validate the different layers of the adaptive system. On the other hand, we will use the two "generations" of the system to validate the framework. The validity study will address both the granularity of the layers, and the proposed per-layer evaluation criteria, methods and instruments.

4 Summary and conclusions

This paper has presented the Prospector system, an adaptive front-end to the Google search engine, which re-ranks results according to user- an group- interests, identified and represented according to ODP-based thematic taxonomies. We have tried to follow an as simple as possible approach both in modeling and in effecting adaptivity. The primary aim in doing so has been to make Prospector as portable as possible across different search engines, or search frameworks in general, with the overarching goal being to develop a generalised adaptive front-end for personalized searching.

Preliminary informal evaluation activities have provided encouraging results and valuable input for continued work on the system. We plan to apply the improvements outlined herein, and engage in a full-scale empirical evaluation of the system, using the layered evaluation approach.

References

[Brin and Page, 1998] S. Brin, and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th World Wide Web Conference*, Brisbane, Australia, April 1998. *Computer Networks and ISDN Systems*, 30(1-7): 107-117, 1998.

[Chirita *et al.*, 2005] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using ODP Metadata to Personalize Search. In *Proceedings of the 28th ACM International SIGIR Conference on Research and Development in Information Retrieval*, pages Salvador, Brazil, August 2005. ACM.

[Jameson, 2003] A. Jameson. Adaptive Interfaces and Agents. In *Human-computer interaction handbook*, pages 305–330. Erlbaum, Mahwah, NJ, 2003.

[Kay, 2000] J. Kay. Stereotypes, Student Models and Scrutability. In *Proceedings of the 5th international Conference on intelligent Tutoring Systems*, pages 19-30, Montréal, Canada, June 2000 (Lecture Notes In Computer Science, vol. 1839). Springer-Verlag, Berlin.

[Koychev and Schwab, 2000] I. Koychev, and I. Schwab. Adaptation to Drifting User's Intersects. In *Proceedings of ECML2000/MLnet workshop "Machine Learning in the New Information Age"*, pages 39-45, Barcelona, Spain, May-June 2000.

[Micarelli and Sciarrone, 2004] A. Micarelli, and F. Sciarrone. Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. *User Modeling and User-Adapted Interaction*, 14(2-3): 159-200, 2004.

[Paramythis and Weibelzahl, 2005] A. Paramythis, and S. Weibelzahl. A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In *Proceedings of the 10th International Conference on User Modeling* (UM2005), pages 438-442, Edinburgh, Scotland, UK, July 2005 (Lecture Notes in Computer Science LNAI 3538, Springer Verlag). Springer-Verlag, Berlin.

[Pierrakos *et al.*, 2003] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction*, 13(4): 311-372, 2003.

[Smyth *et al.*, 2003a] B. Smyth, E. Balfe, P. Briggs, M. Coyle, and J. Freyne. Collaborative Web Search. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (IJCAI-03), pages 1417-1419, Acapulco, Mexico, August 2003. Morgan Kaufmann.

[Smyth *et al.*, 2003b] B. Smyth, J. Freyne, M. Coyle, P. Briggs, and E. Balfe. I-SPY: Anonymous, Community-Based Personalization by Collaborative Web Search. In *Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 367-380, Cambridge, UK, December 2003. Springer.

[Tanudjaja and Mui, 2002] F. Tanudjaja, and L. Mui. Persona: A Contextualized and Personalized Web Search. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 67 (9), Hilton Waikoloa Village, Island of Hawaii, January 2002. IEEE Computer Society.